DOCUMENT RESUME

ED 278 375 IR 012 486

AUTHOR Schultz, Charles P.
TITLE PIR-1 and PIRPL. A Project in Robotics Education. Revised.
PUB DATE Feb 86
NOTE 25p.; Paper presented at the Florida Instructional Computing Conference 1986. Report on a student project in a master's program at Barry University (Florida).
PUB TYPE Guides - General (050) -- Audiovisual Materials (100) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Science Education; *Computer Software; Diagrams; Higher Education; *Microcomputers; Production Techniques; *Programing Languages; *Robotics
IDENTIFIERS Apple (Computer); Barry University FL; BASIC Programing Language; Commodore VIC 20

ABSTRACT

This paper presents the results of a project in robotics education that included: (1) designing a mobile robot--the Personal Instructional Robot-1 (PIR-1); (2) providing a guide to the purchase and assembly of necessary parts; (3) providing a way to interface the robot with common classroom microcomputers; and (4) providing a language by which the robot could be controlled--Personal Instructional Robot Programming Language (PIRPL). Schematic diagrams accompany a detailed description of the robot's hardware design, and information on the selection of robot components includes a list of the parts required with the vendor, ordering number, and cost for each part. Step-by-step instructions for assembling the robot include a base layout diagram. Instructions are also provided for interfacing the robot with either of two microcomputers--the VIC-20 Commodore and the Apple. The control language--PIRPL--is described, including commands, error messages, and listings of the computer codes for both microcomputers. Diagrams provided include the typical arrangement of a robotic system, and interface cabling and wiring. Additional robot component sources and references complete this report. (DJR)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED278375

PIR-1 and PIRPL

a project in robotics education

by Charles P. Schultz

CED 710

Barry University

Miami, FL

completed December, 1985

revised February, 1986

IR012456

BEST COPY AVAILABLE

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Charles P. Schultz

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

INTRODUCTION

This is a presentation of a project done while completing a Masters of Science degree in Computer Education at Barry University, Miami, Florida, under the direction and guidance of Joel Levine and Dennis Murphy.

The goals of the project were as follows:

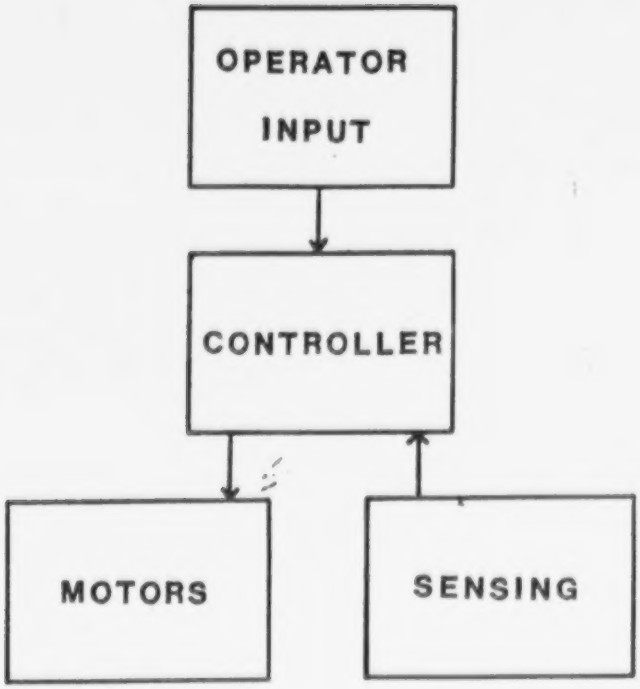
1) Design a mobile robot with sensing capability that could be constructed from common electronic/electromechanical components.

2) Provide a guide to the purchase of the necessary parts, and to the assembly of the robot from these parts.

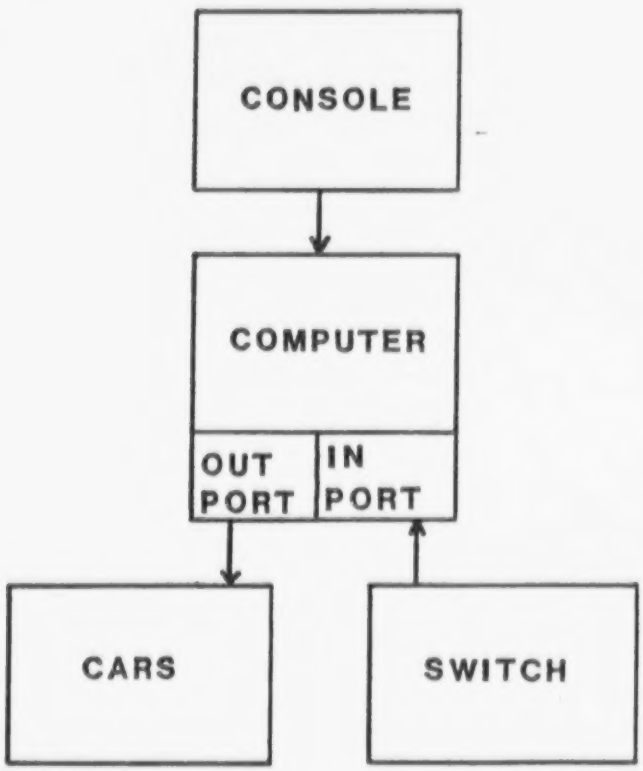
3) Provide a way to interface the robot to common classroom computers.

4) Provide a language by which the robot could be controlled.

These goals were met, and are described and presented in this report.



Typical arrangement of robotic system



Arrangement of PIR-1 system

ROBOT HARDWARE DESIGN

The robot, dubbed Personal Instructional Robot - 1, or PIR-1 (pronounced "pie are one"), was designed to use two inexpensive battery operated cars that would be controlled by a computer through four relays which would cause either car to go forward or backward as desired.

A roller switch on the front of the robot would be a sensing mechanism that would allow detection of a collision between PIR-1 and some object in front of it, thus permitting "artificial intelligence" programming.

Optionally available (not for Apple computers) is a sound effects printed circuit board with flashing led's and accompanying speaker.

Rather than designing the robot to be dedicated to a particular type of computer, it has its own connector which allows "custom" interfacing.

A 74LS367 IC is used to boost the low current signals coming from the controlling computer so they can activate the relay contacts. The relays are single-pole, double-throw (SPDT) reed relays that are activated by a +5 volt signal, thus they can be switched by computer logic level signals.

The robot moves by activating the relays connected to the car motors. The relay contact acts as a switch, applying the battery voltage to one side of the motor, causing it to move the car. A signal set to logic level 1 (approximately +5 volts) closes the contact. If the contact on the plus side of the motor is closed, the car will go forward. If the negative side contact is closed, the car will go backward.

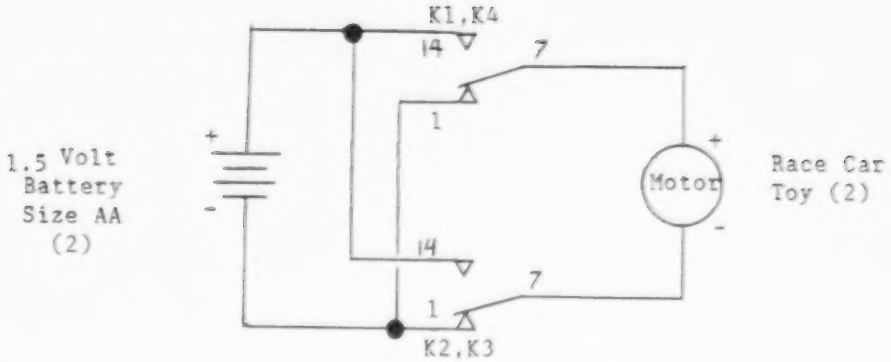
For stability, the cars are mounted in opposite directions, so to make PIR-1 go forward or backward, the cars must go in opposite directions. To turn the robot, the cars must go in the same direction.

The contact sensing switch is attached to a signal that is normally at logic level 1 by being connected to +5 volts through a 10K ohm resistor. When contact is made between the switch and some solid object, the signal is connected through the switch to ground, which is logic level 0.

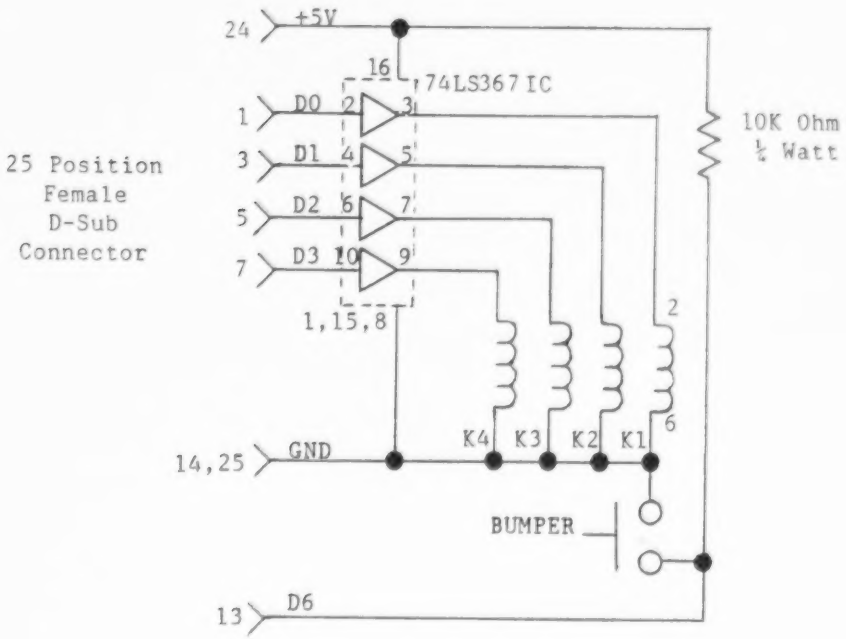
To activate the space board, its signal S1 must be toggled, that is, raised to logic 1 and then brought back to logic 0. After that, activating signal S2 will cause the led's to flash and the space gun sound to occur until S2 is returned to logic 0.

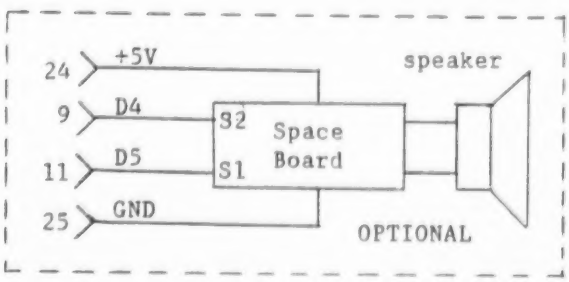
A schematic diagram of the robot electronic hardware is on the next page.

SCHEMATIC DIAGRAMS



K1,K2,K3,K4 = +5VDC SPDT Reed Relays





ROBOT COMPONENT SELECTION

Parts were chosen so that only three sources need to be contacted to purchase all of the necessary parts. Assembly tools required are common, and the only assembly skills necessary are using a hobby knife, glueing, wire preparation, and soldering.

A list of parts required, vendor, ordering number, and cost for each part is given on the next page. The vendors are coded as follows:

DK = DIGI-KEY Corporation
701 Brooks Ave. South
P.O. Box 677
Thief River Falls, MN 56701
1-800-344-4539

HT = Hi Tek Sales
119r Foster Street
Peabody, MA 01961-3357
(617) 532-2323

RS = Radio Shack
contact neighborhood store

ROBOT PARTS LIST

Part	Qty	Vendor	Vendor #	Pricing
Race car	2	HT	85D0264	$ 1.00 ea $ 5.00/ 6
AA Battery	2	HT(NiCad)	85V0360	$ 0.69 ea $ 5.00/10
		RS(Alkaline)	23-552	$ 2.99/ 4
		RS(NiCad)	23-125	$ 3.99/ 4
Battery holder	2	RS	270-401	$ 0.39 ea
5 Volt SPDT Reed Relays	4	DK	Z619-ND	$ 4.40 ea $36.90/10
74LS367IC	1	DK	74LS367ND	$ 0.68 ea
Female DB-25 Connector	1	DK	1S-25S-ND	$ 3.16 ea $25.87/10
		RS	276-1548	$ 3.99 ea
Roller switch	1	RS	275-017	$ 1.29 ea
10K ohm resistor	1	RS	271-1335	$ 0.39/ 5
		DK	10KQ	$ 0.25/ 5
Stiff cardboard (8"x8" min) Space Board (optional)	1	HT	3V0036	$ 4.88 ea $13.00/ 3

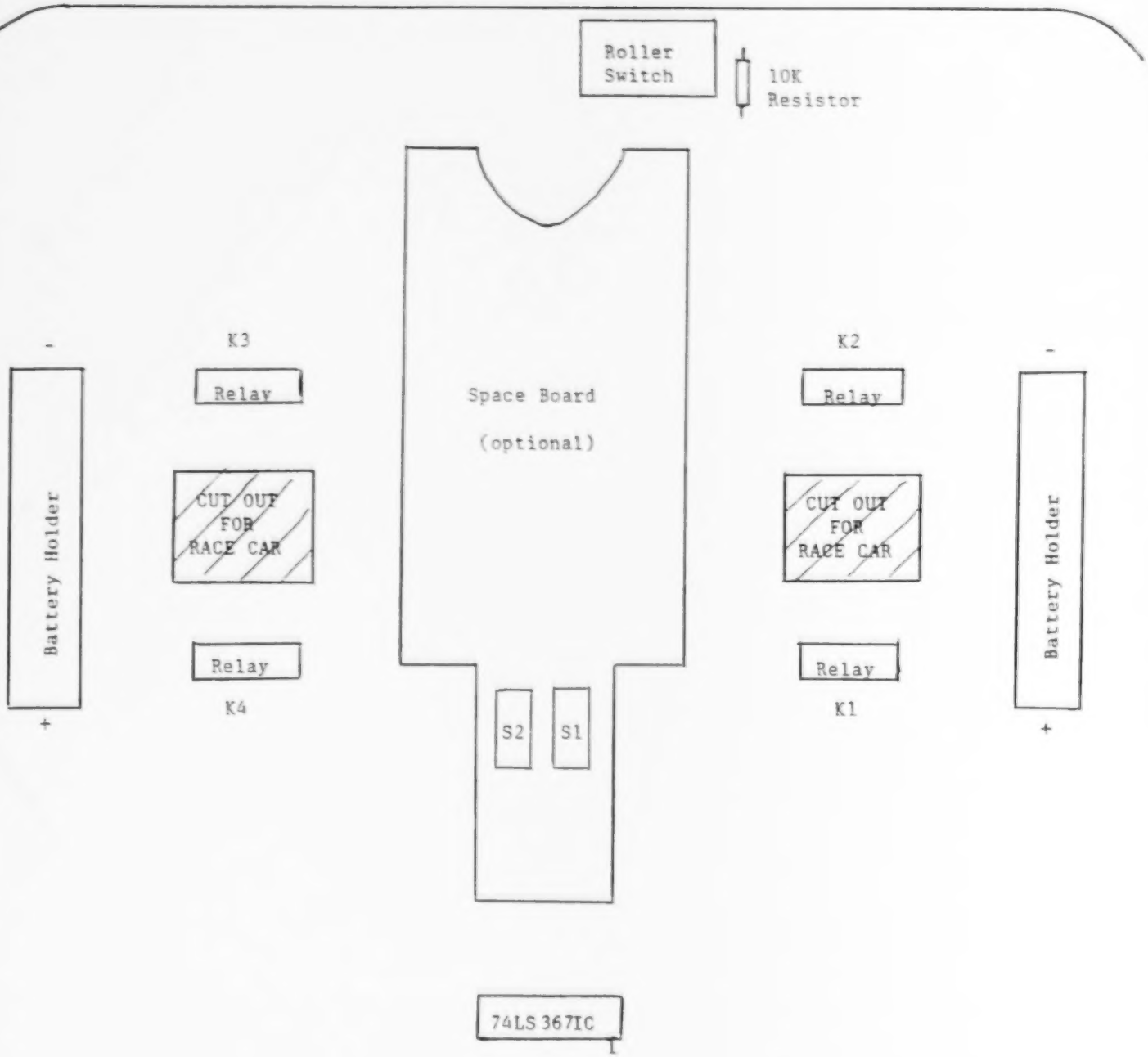
INTERFACE CABLE PARTS LIST

25 Conductor ribbon cable	10'	HT	4BS0594	$ 6.00/10' $20.00/50'
Male DB-25 Connector	1	HT	85BS0149	$ 1.25 ea $10.00/ 9
		RS	276-1547	$ 2.99 ea
		DK	1S-25P-ND	$ 2.42 ea $19.80/10
For VIC-20, Commodore 64:				
24 Position Edge Connector	1	DK	C1-12	$ 2.90 ea $27.50/10
For Apple II+, IIe:				
16 Position Male Dip Header	1	DK	R8103-ND	$ 0.92 ea $ 8.14/10

CONSTRUCTION MATERIALS LIST

Hobby Knife	1	RS	64-1805	$ 0.99 ea
			64-1868	$ 0.99 ea
All-Purpose Adhesive	1	RS	64-2307	$ 1.49 ea
Wire Cutter	1	RS	64-1833	$ 2.99 ea
Wire Stripper	1	RS	64-2129	$ 1.99 ea
Soldering Iron	1	RS	64-2070	$ 4.95 ea
Soldering Iron Stand	1	RS	64-2078	$ 5.79 ea
Roll of Solder	1	RS	64-005	$ 2.99 ea

ROBOT BASE LAYOUT

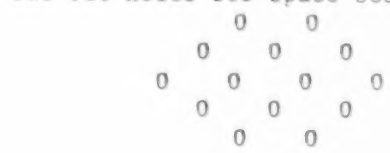


ROBOT TOP LAYOUT

Mount DB-25
Connector here

FOLD

Cut out holes for space board speaker



FOLD

Cut out holes for space board LED's

Cut slot where necessary for roller switch

ROBOT ASSEMBLY INSTRUCTIONS

1) Cut out cardboard base and top using the the templates provided or your own design. Paint and decorate now if you wish to do so.

2) Mount battery holders, relays, 74LS367IC and roller switch to base using all-purpose adhesive. Mount space board speaker to top.

3) Separate the 25 conductor ribbon cable into 16 and 9 conductor lengths. The hook-up wire from the robot will be taken from the 9 conductor length. The 16 conductor length will be used for the interface cable.

4) Solder 4" lengths of wire to the + and - terminals of each race car.

5) Solder wires from the DB-25 connector to the 74LS367as per the schematic.

6) Connect +5V and GND from the DB-25 to the 74LS367per the schematic.

7) Connect the outputs of the 74LS367to the relays according to the schematic.

8) Connect GND from the DB-25 to the relays.

9) Connect the 10K resistor and pin 13 of the DB-25 to the common lead of the roller switch.

10) Connect +5V from the DB-25 to the other end of the resistor.

11) Connect the normally open side of the switch to GND.

12) Bring the car wires through the base cut-outs, and place the base on the cars. Note that the car connected to relays K1 and K2 should be facing the front of the base, while the other car should be facing the opposite direction.

13) Connect the wires from the cars to the relays per the schematic.

14) If the space board is to be used, de-solder the battery connector on the board, connect +5V and GND to where the battery was, and connect S1 and S2 to the DB-25 per the schematic.

15) Mount the DB-25 connector to the robot top using the adhesive.

16) Mount the top to the base after connecting the space board speaker.

17) Assemble the appropriate interface cable according to the information to follow. A female DB-25 connector is used to mate to PIR-1.

18) Be sure to check for shorts between battery terminals and from +5V to GND. If any exist, correct the wiring mistake(s). When the wiring is correct, plug in the robot's batteries, connect the computer and GO!

INTERFACING DESIGN

Interfacing PIR-1 to a computer requires only the basic computer configuration. Thus, each computer in a classroom could be used to control the robot, not, for example, only those with a parallel or RS-232 interface board.

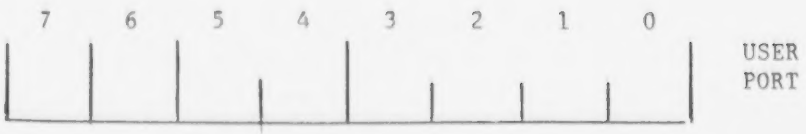
For the VIC-20/Commodore 64 interface, the User Port was chosen to provide the control signals to PIR-1. This port is accessed in BASIC by PEEKing or POKEing and reading or setting the appropriate bit(s). Details and bit assignments are given on the next page.

The Apple interface functions quite differently, using the Game I/O Connector on the motherboard. Because only four output bits are available, only the robot's motors can be controlled. Two additional bits would be required to control the space board, therefore it cannot be used. Also, this connector is normally used for joysticks, so, without additional wiring and tinkering with the joystick interface, the joystick control mode in PIRPL is not available for the Apple.

To use the Game I/O Connector for output, each bit is turned on or off by referencing a different address for each bit, as illustrated on the next page. The reference can be made by a POKE to the appropriate address. To read the contact switch, a PEEK is used in the normal manner.

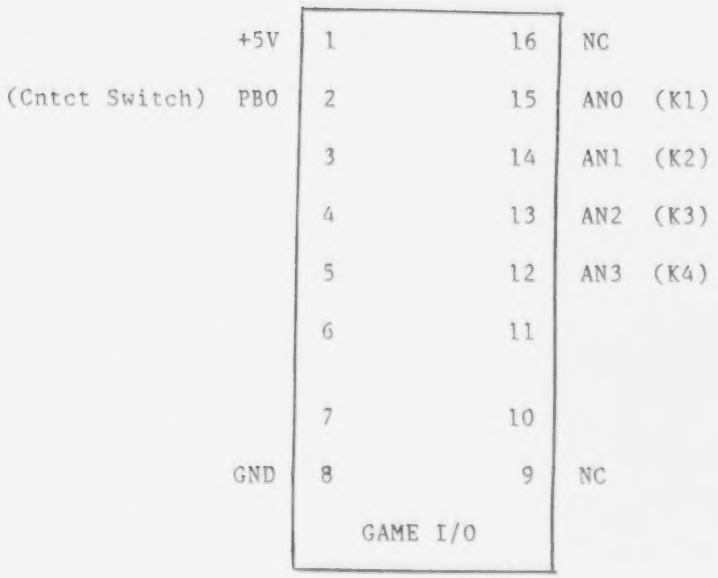
Interface cabling is described on the page following the next. Interfaces for other computers can be developed using the information in this report.

VIC-20/COMMODORE 64 INTERFACING



ADDRESS:
VIC-20 = 37136
COMMODORE 64 = 56577

APPLE INTERFACING



ADDRESS:

AN0 Off	= 49240	AN3 Off	= 49246
On	49241	On	= 49247
AN1 Off	= 49242		
On	49243		
AN2 Off	= 49244	PB0	= 49249
On	49245		

INTERFACE CABLE WIRING

For VIC-20 and Commodore 64:

PIR-1	USER PORT
1	C
3	D
5	E
7	F
9	H
11	J
13	K
14	A
24	2
25	1

For Apple:

PIR-1	GAME I/O
1	15
3	14
5	13
7	12
13	2
14	8
24	1
25	8

LANGUAGE DESIGN

The Personal Instructional Robot Programming Language, or PIRPL (pronounced "purple") was developed to control PIR-1's motion and sensing capabilities. Also included are program control commands and commands to put the robot under direct control of the computer keyboard or a joystick (joystick control not available for Apple computers). The motion control commands imitate LOGO turtle movement commands.

The PIRPL interpreter is written in BASIC, in a form that is easily transportable from one computer type to another. There is a common section of code which is for use on VIC-20, Commodore 64, and Apple computers. Then, computer-dependent subroutines must be added to complete the software package.

If computers other than those for which PIRPL was designed to be used for are desired, then enter computer-dependent routines with the instructions necessary to control the robot with that computer.

Also, if it is desirable to use motion commands other than FD, BK, LT and RT to illustrate a particular principal (eg CW and CCW instead of RT and LT respectively), make the appropriate string replacements in subroutine 100.

Modal Commands:

When using PIRPL, the robot is controlled by commands in response to the 'P?' prompt. By issuing the following commands, the robot's movements can be directly controlled by the specified device.

JOYSTK — causes the robot to be controlled by the movement of a joystick. Pressing the fire button on the joystick produces the same result as the 'FIRE' command (see below). Control is returned to PIRPL by hitting any key on the keyboard. (This feature is not available for the Apple version because the robot interface uses signals required for the joystick.)

KEYBD — causes the robot to be controlled by pressing various keys on the keyboard. I=forward, J=left, K=right, M=back, F=fire, 'space bar'=stop, and Q=quit and return to PIRPL.

Movement Commands:

The follwing commands control movement of the robot. The may be combined on one line, and may be used in conjunction with the 'REPEAT' command (see below).

FD n — causes the robot to move forward 'n' units. One unit equals approximately one inch. A space is required between the command and the number of units to be moved, as well as between instructions.

BK n — causes the robot to move backward 'n' units. Definition of a unit, and syntax are the same as for 'FD' above.

LT n — causes the robot to turn to the left approximately 'n' degrees. Syntax is the same as for 'FD' above.

RT n — causes the robot to turn to the right approximately 'n' degrees. Syntax is the same as for 'FD' above.

Other commands:

The following commands control auxillary and sensing functions of the robot, as well as conditional and repeated execution of other PIRPL commands.

FIRE — causes robot built with optional 'space module' to blink led's and make a laser firing noise.

IFTOUCH (...) — causes instruction(s) within parentheses to be executed only when the switch on the front of the robot has made contact with some object (person, wall, etc.). A space is required between the 'IFTOUCH' and the left parenthesis, as well as between the left parenthesis and the first command inside the parentheses. Also put a space after the closing parenthesis for a command outside the parentheses.

REPEAT n[...]
causes instruction(s) within brackets to be executed 'n' number of times. A space is required between 'REPEAT' and the number of times to repeat, but there cannot be a space between the repeat factor and the left bracket. Also, there must be a space between the left bracket and the first instruction within brackets, and between the right bracket and any command outside the brackets.

More Commands:

By making additional sensing or program control capabilities you can customize PIRPL for your specific application. Also, as stated before, existing PIRPL commands can be changed to meet your needs.

PIRPL ERROR MESSAGES

* MISSING (*

An instruction requiring a left parenthesis does not have one.

* MISSING) *

An instruction requiring a right parenthesis does not have one.

* MISSING [*

An instruction requiring a left bracket does not have one.
Also occurs when adding a space between the repeat factor and the bracket for the REPEAT command, or when omitting the space between the left bracket and the first command inside brackets.

* MISSING] *

An instruction requiring a right bracket does not have one.

* SYNTAX ERROR *

An instruction is not recognized, one or more instruction parameters are missing, or there is more than one space between commands and data.

PIRPL PROGRAM: COMMON PORTION

```
1 REM *PIRPL BY C. SCHULTZ *
2 REM *VERSION 1.3*
5 GOSUB 1500
10 GOSUB 2400
15 GOSUB 1000
20 GOSUB 1200
30 PRINT : PRINT "P";
40 INPUT A$
50 ES=0
55 TF=0
60 Q=0
70 RPT=0
80 IF A$="" THEN 30
90 GOSUB 100
92 IF RPT=1 THEN 300
95 IF Q=0 THEN 30
97 GOSUB 1300
99 END

100 REM *PARSE STRING*
101 GOSUB 200
103 IF SA$="QUIT" THEN Q=1 : GOTO 195
105 IF SA$="IFTOUCH" THEN GOSUB 800 : RETURN
106 IF SA$="FIRE" THEN GOSUB 2600 : RETURN
107 IF SA$="REPEAT" THEN RPT=1 : RETURN
108 IF SA$="KEYBD" THEN GOSUB 700 : RETURN
109 IF SA$="JOYSTK" THEN GOSUB 600 : RETURN
110 IF SA$="FD" THEN GOSUB 2000 : SF=100 : GOTO 150
120 IF SA$="BK" THEN GOSUB 2100 : SF=100 : GOTO 150
130 IF SA$="LT" THEN GOSUB 2200 : SF=6 : GOTO 150
140 IF SA$="RT" THEN GOSUB 2300 : SF=6 : GOTO 150
145 PRINT " * SYNTAX ERROR *"
149 RETURN
150 GOSUB 200
160 IF VAL(SA$)=0 THEN PRINT " * SYNTAX ERROR *" : GOTO 195
170 FOR I=1 TO SF*VAL(SA$) : NEXT I
180 GOSUB 2400
190 IF ES<>1 THEN 100
195 RETURN

200 REM *GET SUBSTRING*
205 PTR=1 : ES=0
210 IF MID$(A$,PTR,1)=" " THEN 250
220 PTR=PTR+1
230 IF PTR>LEN(A$) THEN ES=1 : GOTO 250
240 GOTO 210
250 SA$=MID$(A$,1,PTR-1)
260 IF ES=1 THEN 290
270 A$=MID$(A$,PTR+1,LEN(A$))
290 RETURN
```

PIRPL PROGRAM: COMMON PORTION (continued)

```
300 REM *FIND BRACKET*
305 GOSUB 200
310 IF MID$(SA$,LEN(SA$),1)<>"[" THEN PRINT " * MISSING [ *" : GOTO
    GOTO 30
315 B$=SA$ : GOSUB 500 : SA$=B$
317 IF TF=1 THEN RF=1 : GOTO 340
320 RF=VAL(MID$(SA$,1,LEN(SA$)-1)
325 IF RF=0 THEN PRINT " * SYNTAX ERROR *" : GOTO 30
330 GOSUB 400
335 IF ER=1 THEN 30
340 FOR J=1 TO RF
350 A$=RA$
360 GOSUB 100
370 NEXT J
380 A$=TA$
390 GOTO 70

400 REM *GET STRING IN []*
405 PTR=1
410 IF MID$(A$,PTR,1)="]" THEN 450
420 PTR=PTR+1
430 IF PTR LEN(A$) THEN ER=1 : GOTO 490
440 GOTO 410
450 RA$=MID$(A$,1,PTR-1)
460 B$=RA$ : GOSUB 500 : RA$=B$
470 IF ER=1 THEN 495
480 TA$=MID$(A$,PTR+2,LEN(A$))
490 IF ER=1 THEN PRINT " * MISSING ] *"
495 RETURN

500 REM *TRIM SPACES*
510 IF LEN(B$)<=1 THEN B$="" : PRINT " * SYNTAX ERROR *" : ER=1 :
    RETURN
520 IF MID$(B$,LEN(B$),1)=" " THEN B$=MID$(B$,1,LEN(B$)-1) : GOTO 510
530 RETURN

700 REM *KEYBD*
705 GET A$
710 GOSUB 2500
715 IF TCH=0 THEN GOSUB 2400 : RETURN
720 IF A$="" THEN 705
730 IF A$=" " THEN GOSUB 2400
740 IF A$="Q" THEN GOSUB 2400 : RETURN
750 IF A$="I" THEN GOSUB 2000
760 IF A$="J" THEN GOSUB 2200
770 IF A$="K" THEN GOSUB 2300
780 IF A$="M" THEN GOSUB 2100
790 GOTO 705
785 IF A$="F" THEN GOSUB 2600
```

PIRPL PROGRAM: COMMON PORTION (continued)

```
800 REM *IFTOUCH*
810 GOSUB 200
820 IF SA$<>"(" THEN PRINT " * MISSING ( *" : ER=1 : RETURN
830 TPTR=1
840 IF MID$(A$,TPTR,1)=")" THEN 890
850 TPTR=TPTR+1
860 IF TPTR>LEN(A$) THEN PRINT " * MISSING ) *" : ER=1 : RETURN
880 GOTO 840
890 FA$=MID$(A$,1,TPTR-2)
895 GOSUB 2500
900 IF TCH=0 THEN T$=A$ : A$=FA$ : GOSUB 100
910 A$=T$
920 RETURN

1000 REM *TITLE PAGE*
1010 GOSUB 1200
1020 PRINT : PRINT "    WELCOME TO PIRPL"
1030 PRINT : PRINT
1040 PRINT "  PERSONAL"
1050 PRINT "  INSTRUCTIONAL"
1060 PRINT "  ROBOT"
1070 PRINT "  PROGRAMMING"
1080 PRINT "  LANGUAGE"
1090 PRINT : PRINT : PRINT : PRINT
1100 PRINT "(C) COPYRIGHT 1985"
1110 PRINT "BY C. SCHULTZ"
1120 PRINT : PRINT "HIT RETURN TO CONTINUE"
1130 GET A$ : IF A$="" THEN 1130
1140 RETURN

1300 REM *GOODBYE*
1310 GOSUB 1200
1320 PRINT "THANKS FOR PLAYING"
1330 PRINT "WITH ME!"
1340 PRINT
1350 RETURN


end of common portion of PIRPL
```

PIRPL PROGRAM: APPLE PORTION

```
600 REM *JOYSTK*
610 PRINT "JOYSTK NOT AVAILABLE"
620 RETURN

1200 REM *CLEAR SCREEN*
1210 HOME
1220 RETURN

1500 REM *INITIALIZE*
1510 A0=49240
1520 A1=A0+2
1530 A2=A1+2
1540 A3=A2+2
1550 TC=49249
1560 RETURN

2000 REM *FORWARD*
2010 POKE A0+1,0
2020 POKE A1,0
2030 POKE A2+1,0
2040 POKE A3,0
2050 RETURN
2100 REM *BACKWARD*
2110 POKE A0,0
2120 POKE A1+1,0
2130 POKE A2,0
2140 POKE A3+1,0
2150 RETURN
2200 REM *LEFT*
2210 POKE A0,0
2220 POKE A1+1,0
2230 POKE A2+1,0
2240 POKE A3,0
2250 RETURN
2300 REM *RIGHT*
2310 POKE A0+1,0
2320 POKE A1,0
2330 POKE A2,0
2340 POKE A3+1,0
2350 RETURN
2400 REM *STOP*
2410 POKE A0,0
2420 POKE A1,0
2430 POKE A2,0
2440 POKE A3,0
2450 RETURN

2500 REM *CHECK TOUCH*
2510 TCH=1
2520 IF PEEK(TC)>127 THEN
     TCH=0
2530 RETURN

2600 REM *FIRE*
2610 PRINT "FIRE NOT AVAILABLE"
2620 RETURN
```

PIRPL PROGRAM: VIC-20/COMMODORE 64 PORTION

```
600 REM *JOYSTK*
610 POKE DD,127 :REM for VIC-20 only
615 PJ=PEEK(JR)
620 IF (PJ AND UP)=0 THEN GOSUB 2000 : GOTO 670
630 IF (PJ AND DOWN)=0 THEN GOSUB 2100 : GOTO 670
640 IF (PJ AND LEFT)=0 THEN GOSUB 2200 : GOTO 670
650 IF (PEEK(PK) AND RIGHT)=0 THEN GOSUB 2300 : GOTO 670
660 IF (PJ AND FIRE)=0 THEN GOSUB 2600 : GOTO 670
665 GOSUB 2400
670 GOSUB 2500
675 IF TCH=0 THEN 690
680 GET A$ : IF A$="" THEN 615
690 GOSUB 2400
692 POKE DD,255 :REM for VIC-20 only
695 RETURN

1200 REM *CLEAR SCREEN*
1210 PRINT CHR$(147)
1220 RETIRN

1500 REM *INITIALIZE* for VIC-20 only
1510 RC=37136
1520 JR=37137
1530 POKE 37138,63
1540 POKE 37139,0
1550 POKE RC,32
1560 DD=37154
1570 PK=37152
1580 UP=4 : DOWN=8 : LEFT=16 : RIGHT=128 : FIRE=32

1500 REM *INITIALIZE* for COMMODORE 64 only
1510 RC=56577
1520 JR=56320
1530 POKE 56579,63
1550 POKE RC,32
1570 PK=56320
1580 UP=1 : DOWN=2 : LEFT=4 : RIGHT=8 : FIRE=16

1590 RETURN

2000 REM *FORWARD*
2010 POKE RC,5
2020 RETURN
2100 REM *BACKWARD*
2110 POKE RC,10
2120 RETURN
2200 REM *LEFT*
2210 POKE RC,9
2220 RETURN
2300 REM *RIGHT*
2310 POKE RC,6
2320 RETURN
2400 REM *STOP*
2410 POKE RC,0
2420 RETURN

2500 REM *CHECK TOUCH*
2510 TCH=(PEEK(RC) AND 64)
2520 RETURN

2600 REM *FIRE*
2610 POKE RC,16
2620 FOR K=1 TO 300
2630 NEXT K
2640 RETURN
2635 POKE RC,0
```

ADDITIONAL ROBOT COMPONENT SOURCES

Since the original writing of this report, other sources for components have been identified. They are:

AE = All Electronics Corporation
905 S Vermont Avenue
P.O. Box 20406
Los Angeles, CA 90006
1-800-826-5432

JE = JAMECO Electronics
1355 Shoreway Road
Belmont, CA 94002
(415) 592-8097

Part	Qty	Vendor	Vendor #	Pricing
5 Volt SPDT Reed Relays	4	AE	RRLY-DIP5	$ 1.75/ea $16.00/10
74LS367IC	1	AE	74LS367	$ 0.65 ea
		JE	74LS 367	$ 0.65 ea
Female DB-25 Connector	1	AE	DB25S	$ 2.95 ea $25.00/10
		JE	DB25S	$ 2.95 ea
Roller Switch	1	AE	SMS-350	$ 0.75 ea
For VIC-20, Commodore 64:				
24 Position Edge Connector	1	JE	12/24SE	$ 3.95 ea $34.90/10
For Apple II+, IIe:				
16 Position Male Dip Headr	1	AE	16HIC	$ 0.55 ea
		JE	16PINHP	$ 0.69 ea $ 6.50/10

REFERENCES

All Electronics Corp.
Fall 1985 Catalog #385

JAMECO Electronics
1985 Catalog

REFERENCES

VIC 20 Programmer's Reference Guide
Finkel, Harris, Higginbottom, and Tomczyk
Commodore Business Machines, Inc. and
Howard W. Sams & Co., Inc.

VIC Revealed
Nick Hampshire
Hayden Book Co., Inc.

Commodore 64 Programmer's Reference Guide
Commodore Business Machines, Inc. and
Howard W. Sams & Co., Inc.

Mapping the Commodore 64
Sheldon Leemon
COMPUTE! Publications, Inc.

Applesoft II BASIC Programming Reference Manual
Apple Computer Inc.

Apple II Reference Manual
Apple Computer Inc.

HiTek Sales Retail-Wholesale Catalog
#85H3

Digi-Key Corporation Catalog
#855 Sept.-Oct., 1985

Radio Shack 1986 Catalog
#393

Apple is a registered trademark of Apple Computer Inc.

The words BASIC and LOGO used in this report refer to computer languages, but not any particular version of these languages.